독성 감지를 위한 생물 조기 경보 시스템

김성용* · 권기용* · 이원돈**

Biological Early Warning System for Toxicity Detection

Sung Yong Kim* · Ki Yong Kwon* · Won Don Lee**

이 논문은 2009년도 충남대학교 학술연구비에 의해 지원되었음

요 약

생물 조기 경보 시스템은 물속 생명체의 행동을 관찰하여 독성을 감지한다. 이 시스템은 분류기를 물의 독성의 유무와 정도를 판단하기 위해 사용한다. 이 분류기의 성능을 높이기 위해 적용할 수 있는 방법 중에 부스팅 알고리즘이 있다. 부스팅은 기본 분류기로는 예측 정확도가 낮았던 분류하기 어려운 사건에 집중할 수 있도록 다음 번 데이터에 해당 훈련 사건(event)들이 뽑힐 확률을 높여준다. 횟수가 진행될수록 분류기가 어려운 사건들을 집중적으로 고려하게 된다. 그 결과 분류하기 어려웠던 사건에 대한 예측 성능은 좋아지지만, 비교적 쉬운 훈련 사건들의 정보는 버려지는 단점이 있다. 본 논문에서는 이 같은 단점을 보완하기 위해 분류기에 확장된 데이터 표현을 위한 점진적 학습법의 적용을 제안한다. 확장된 데이터 표현의 가중치 변수를 사용하면 약하게 분류되는 사건 뿐 아니라 쉽게 분류되는 사건의 정보까지도 사용하여 분류기의 예측 정확도를 높일 수 있게 된다. 새로 적용된 알고리즘과 기존의 중요도 변수를 사용하지 않는 learn++를 비교하여 성능이 향상됨을 검증하였다.

ABSTRACT

Biological early warning system detects toxicity by looking at behavior of organisms in water. The system uses classifier for judgement about existence and amount of toxicity in water. Boosting algorithm is one of possible application method for improving performance in a classifier. Boosting repetitively change training example set by focusing on difficult examples in basic classifier. As a result, prediction performance is improved for the events which are difficult to classify, but the information contained in the events which can be easily classified are discarded. In this paper, an incremental learning method to overcome this shortcoming is proposed by using the extended data expression. In this algorithm, decision tree classifier define class distribution information using the weight parameter in the extended data expression by exploiting the necessary information not only from the well classified, but also from the weakly classified events. Experimental results show that the new algorithm outperforms the former Learn++ method without using the weight parameter.

키워드

부스팅, 점진적 학습, 의사결정 트리, 확장된 데이터 표현, 엔트로피 함수

Key word

Boosting, Incremental learning, Decision tree, Extended data expression, Entropy function

* 충남대학교 컴퓨터공학과
** 충남대학교 컴퓨터공학과 (교신저자, wlee@cnu.ac.kr)

접수일자 : 2010. 05. 19
심사완료일자 : 2010. 08. 16

Ⅰ. 서 론

수질을 파악할 때 사용하는 기존의 화학적인 방법은 샘플을 수집하여 분석하는 방법이었다. 이 방법은 샘플을 수집하는 시기의 수질 정보만 알 수 있으며 사용자가 정의하지 않은 물질을 탐지해 낼 수 없는 방법론적인 한계가 있다. 이에 반해 생물 조기 경보 시스템(Biological Early Warning System, BEWS[1][2][3])은 수질 내 유기물 전체의 생물적 반응을 지속적으로 감시하여 독성을 탐지한다. 따라서 수질 상태를 실시간으로 파악할 수 있고, 기존의 화학적 방법으로 발견하지 못했던 독성을 감지할 수 있는 등 많은 장점이 있다. 이러한 BEWS가 성공적으로 적용되기 위해서는 수질의 독성 같은 오염 발생을 신뢰성 있게 예측할 수 있어야 한다.

기존 BEWS[4][5]에서 예측에 사용되는 일반적인 분류기(classifier)는 단일 분류기가 말단 노드의 대표 클래스를 선택하기 때문에 대표로 선택되지 못한 클래스를 포함하는 사건들을 잘 분류하지 못한다는 단점이 있다. 이와 달리 부트스트랩(bootstrap)[6]을 기초로 제안된 알고리즘은 각각의 데이터의 집합으로부터 (교체를 수행하면서) 반복적인 추출(sampling)을 수행하는 기법이다. 각각의 학습 데이터 집합들로부터 약한 분류기들을 생성하고, 생성된 분류기들의 앙상블을 통하여 강한 분류기를 만들어 낸다.

앙상블 기반 시스템을 사용하는 경우, 두 가지 중요한 요소가 있다. 첫 번째 중요한 요소는 다양한 분류기들의 앙상블을 만들어 내는 것이다. 부스팅(boosting) [7][8]은 기본 분류기에서 잘 분류하지 못했던 어려운 사건에 가중치를 두어 학습 데이터 집합을 추출한다. 가중치가 강화된 사건은 다음 회의 데이터 갱신에서 분류기를 학습할 학습 데이터 집합에 들어갈 확률이 높다. 각각의 사건들은 학습 데이터 집합에 중복되어 들어갈 수 있다. 이렇게 만들어진 학습 데이터는 분류기가 분류가 어려운 사건들을 더 중요하게 분류할 수 있도록 한다.

두 번째 중요한 요소는 분류기들을 결합하기에 효율적인 알고리즘이라는 것이다. 지난 수년간 부스팅 알고리즘의 여러 방법들이 개발되어 왔다. 이 알고리즘들은 어떠한 방법으로 훈련 사건들의 가중치가 각 부스팅 과정의 끝에 갱신되어지는지, 어떠한 방법으로 각 분류의 예측이 종합되어지는지에 따라 다르다.

이에 관련하여 본 논문은 클래스를 결정하기 위해 가중치된 투표로서 각 클래스가 가지는 확률을 고려하고 약한 분류기들로부터 확장된 가설(hypothesis)들을 받는 확률적 가중치를 이용한 과반수 투표 방법을 제안한다. 이 알고리즘은 절단(prunning)과정을 통해 생성된 말단 노드에 완전히 분류되지 않고 여러 클래스가 혼재되어 있는 상황에서, 가장 높은 확률을 가지는 클래스를 대표 클래스로 선택하는 것 대신에 각각의 클래스가 가지는 확률 정보를 모두 학습 알고리즘에서 고려한다. 따라서 가장 높은 확률을 가지는 클래스가 선택되고 나머지 정보는 버려지는 일반적인 학습방법에 비해 정보를 잃는 것을 피할 수 있다.

본 논문에서는 이를 BEWS에 적용하기 위해 약한 분류기로서는 확장 데이터 표현 방법(extended data expression)을 적용할 수 있는 UChoo[9]를 사용하고 동시에 점진적 학습(incremental learning)에서는 가중치의 변화에 따른 학습을 사용한 UChoo Boost 방법이 적용된 실험을 통하여 기존의 Learn++[6] 방법보다 더 성능이 향상됨을 보인다.

Ⅱ. 선행 연구

1. 변수들

실험에 사용된 BEWS[5]는 주어진 어항에 물고기를 넣고 그 물고기가 행동하는 것을 관찰하여 비정상적인 운동의 경우 경보를 발생시킨다. 아날로그 방식인 물고기의 행동궤적을 디지털화 하기 위하여 그 궤적을 나타낼 수 있는 변수(attribute)들을 정의한다. 변수들은 x좌표, y좌표, 거리, 절대 거리, 각도, 프랙탈 차원으로 6개이다. 그런 변수들로 이루어진 벡터들로 물고기의 행동이 묘사되고 그 벡터들을 사용하여 룰을 만들게 된다. 이러한 변수 중의 하나로 프랙탈이 있다.

어항에 있는 물고기는 정면으로 관찰되어지므로 2차원의 평면위에서 움직이는 궤적을 보인다. 주어진 시간에 얼마만큼이나 활발하게 행동하느냐는 것을 표현하는 데에 프랙탈 변수가 쓰인다. 프랙탈 도형[10]에서 프랙탈 차원 D를 계산하기 위하여 여러 가지 방법이 제안되었는데 그 중에 상자수 계산(box counting), 상관관계(correlation) 차원 등이 일반적으로 널리 사용

된다.

본 실험에서는 프랙탈 변수의 값을 상자 수 계산 방법에 기초하여 산출한다. 이 상자 수 계산 방법은 컴퓨터로 수행하기가 쉽다는 것과 복잡한 이미지들에게도 적용이 가능 하다는 두 가지 중요한 장점을 제공한다. 아래 그림 1에서 보는 것과 같이 물고기가 움직인 궤적이 지나가는 상자 수를 재귀적으로 계산하게 되면 프랙탈 값에 수렴하게 되는 것을 알 수 있다. 만약에 물고기의 행동이 매우 활발하여 주어진 시간에 모든 박스를 다 지나가게 된다면 프랙탈 값은 2차원 평면 상에서 가질 수 있는 이상적인 최대값인 2가 될 것이다. 그러나 대부분의 경우엔 그보다 작은 0과 2 사이의 값을 가지게 된다.

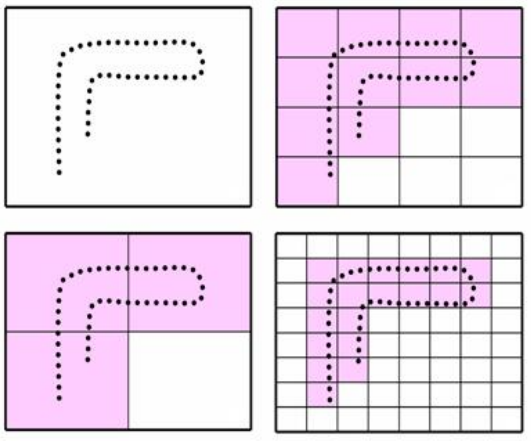

그림 1. 프랙탈 차원
Fig. 1. Fractal Dimension

2. C4.5

분류는 학습(learning)단계에서 입력 데이터의 속성 및 클래스의 관계를 가장 잘 판별할 수 있는 분류기를 찾는다. 이 분류기는 분류기를 만드는데 사용된 데이터는 물론, 처음 보는 사건들의 클래스도 정확하게 예측하여야 한다. 이와 같은 방법에는 의사 결정 트리(decision tree), 규칙기반 분류기(rule-based classifier), 신경회로망(neural networks), 지지도 벡터 기계(support vector machine), 순수 베이지안 분류기(naïve Bayes classifier) 등이 있다.

이 중에서 의사 결정트리를 통한 데이터 분석의 결과는 트리 구조로 표현되기 때문에 분석가가 결과를 쉽게 이해하고 설명할 수 있으며, 전문가가 판단하여 활용할 수 있다는 장점이 있기 때문에 인공지능, 기계학습, 통계분석 등에서 많이 활용되고 있다.

의사 결정 트리에는 다양한 알고리즘들이 사용된다. 그 중 데이터 마이닝에서 많이 언급되고 널리 사용되는 알고리즘은 C4.5[7]이다. C4.5는 최소의 메모리를 사용하고 높은 속도의 성능을 보이는 뛰어난 분류 알고리즘이다. 의사 결정트리는 뿌리 노드(root node) 내부 노드(internal node)와 말단 노드(leaf node)로 구성 되어 있다. 뿌리 노드와 내부 노드에선 변수들 중 엔트로피가 가장 낮은 한 개가 선택된다. C4.5 알고리즘은 각 변수들의 엔트로피를 비교하기 위해 정보 이득(gain_ratio)을 계산한다. 정보 이득은 해당 변수로 분류하였을 경우 클래스가 얼마나 잘 나뉘었는가를 측정하기 위하여 사용한다. 따라서 이들 중 가장 큰 정보 이득를 가진 변수를 선택하여 해당 내부 노드 또는 부모 노드로 결정한다. 위와 같은 방법을 모든 노드에 대해 적용하면 데이터 집합에 따른 의사 결정 트리를 얻을 수 있다.

3. 확장된 데이터 표현

표 1은 일반적으로 C4.5에서 사용되는 데이터를 세분화된 항목으로 변형한 것이다. 각 사건이 가지고 있는 정보에 따라 변수와 클래스의 항목은 0과 1로 표현되어 있다. UChoo에서는 확장된 표현인 표 1을 이용하여 각 항목에 확률적 개념을 도입한다. 표 1의 1번 사건에 대해 Quality는 Good과 Bad에 대해 1/2씩 동일하고, Price가 $45일 때, 살 확률이 3/5 이라는 뜻이다.

또한 각 사건은 가중치 값을 갖게 된다. 이것은 그 사건이 얼마만큼의 중요도(i)를 가지는가를 나타낸다. 일반적인 레코드의 중요도가 1이라고 보았을 때, 가중치 20인 사건은 다른 가중치 1인 레코드의 20개에 해당하는 중요도를 가진 사건이라는 뜻이 된다.

그러므로 여기서 하나의 사건은 동일한 값을 갖는 레코드들의 집합이다. 따라서 전체의 사건의 개수와 레코드들의 개수는 서로 다른 값일 수 있다. 위의 표 1에서 전체 레코드는 24개인데 반해, 전체 사건 개수는 5개이다.

실생활에서 데이터로 변환하여 사용되는 수치에는 확률적인 문제가 세세하게 고려되어야 하는 부분이 많다. 식사에 사용되는 갖가지 양념의 비율 문제는 작은 변화가 최종 결과에 큰 변화를 줄 수 있는 좋은 예이다. 확

표 1. 훈련 데이터의 예
Table. 1 Training data set

사건#	중요도 (i)	Quality		Price			Class	
		Good	Bad	45	30	15	Buy	Don't Buy
1	20	1/2	1/2	1	0	0	3/5	2/5
2	1	0	1	1	0	0	0	1
3	1	0	1	0	1	0	0	1
4	1	1	0	0	1	0	1	0
5	1	0	1	0	0	1	1/2	1/2

장된 데이터 표현을 사용하는 UChoo 분류기는 세분화된 항목과 확률적인 표현으로 기존의 의사결정트리 보다 현실에 근접한 데이터로 규칙(rule)을 만들 수 있을 뿐만 아니라 불완전한 데이터, 손실된 데이터 등을 정의[11] 할 수 있는 장점을 갖는다.

4. 점진적 학습법

실생활에서 사용되고 있는 정보와 관련된 분류 알고리즘은 기존의 방대한 데이터는 물론 실시간으로 수집되는 새로운 데이터 또한 계속적으로 반영할 수 있어야 한다. 기존의 전형적인 분류 알고리즘[7]들은 이러한 작업을 수행할 때, 새로운 데이터와 전의 데이터를 모두 사용하여 분류기를 갱신해 왔다. 이러한 방식은 적은 데이터를 추가하기 위해 큰 기존 데이터를 다시 학습해야 하는 비효율적인 측면 뿐만 아니라, 만약 기존 데이터를 사용할 수 없는 경우, 더 이상 새로운 데이터를 추가하여 갱신을 할 수 없게 된다는 단점이 있다.

이런 문제를 해결할 수 있는 점진적 학습법으로 대표적인 알고리즘 중에 Learn++[6]가 있다. Learn++는 Adaboost[7]나 부스팅처럼 부트스트랩을 기초로 제안된 알고리즘이다. 점진적 학습법은 기존의 데이터로 만들어진 규칙에 새로운 데이터로 만들어진 규칙을 합성하여 분류기를 갱신한다. 만들어진 규칙을 사용하기에 기존의 데이터를 학습할 필요가 없어서 효율적이며, 기존의 데이터를 잃어버릴 경우에도 규칙만 있으면 새로운 데이터로 만든 규칙과 합성하여 분류기를 갱신할 수 있다.

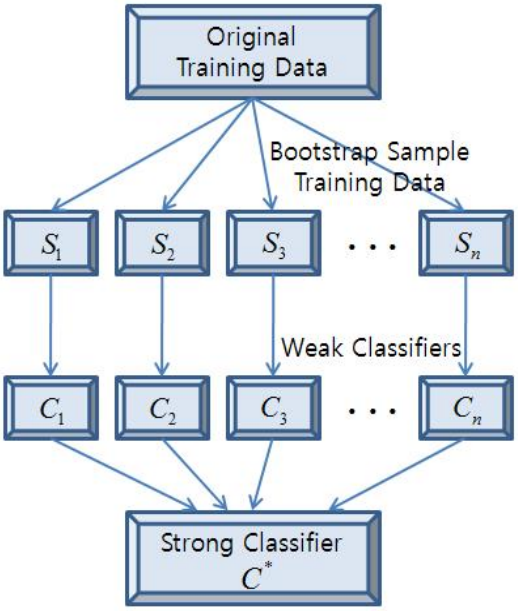

그림 2. 분류기들의 앙상블
Fig. 2. Ensemble of classifiers

5. 확률적 표현에 기반한 UChoo Boost

부트스트랩을 기초로 제안된 알고리즘은 그림 2에서처럼 학습 데이터 집합으로부터 추출된 각각의 데이터 집합들(s)로부터 약한 분류기(c)들을 생성하고 생성된 분류기들의 앙상블을 통하여 강한 분류기(C)를 만들어 낸다.

앙상블 기반 시스템을 사용하는 경우, 부스팅 알고리즘은 어떠한 방법으로 각 분류의 예측이 종합되어지는지에 따라 그 성능과 특징이 달라지게 된다.

기존의 부스팅 알고리즘에서는 먼저 사용될 데이터베이스 분포를 균일하게 초기화한다. 모든 사건들은 동일한 확률로 선택되어 훈련 집합에 포함된다. 약한 분류기는 훈련 사건 집합으로부터 약한 분류기를 통해 가설(hypothesis)을 얻게 된다.

이 가설들은 합쳐져서 분류계산에 사용된다. 현재 약한 분류기의 결과인 가설의 성능에 따라 잘못 분류된 사건들의 가중치가 갱신된다. 다음 단계에서는 이와 같은 가중치를 이용해 다시 훈련 집합을 뽑는다. 이 과정에서 분류하기 어려운 사건들은 가중치가 올라가서 더 잘 선택되게 된다. 그 결과 분류기는 분류하기 어려운 사건들에 더 집중하게 된다. 최종 가설은 데이터베이스의 모든 가설들의 결합으로 얻어진다. 부스팅은 위의 방법으로 기본 분류기가 분류하기 어려운 사건을 집중적으로 다음 번의 갱신에 선택하여 예측 확률을 높이는 방식이다.

단점은 다음 차례(round)에 선택이 되지 못한 사건들에 대한 정보는 분류기가 알 수 없다는 점이다. 예를 들어, 의사 결정 트리의 경우 말단 노드에서 100개의 사건들 중 클래스 1이 70개이고, 클래스 2가 30개라면, 약한 분류기일 경우의 가설은 해당 말단 노드에는 클래스 1을 할당한다. 클래스 2라는 정보를 가지고 있는 30개의 사건은 잘못 측정이 되는 것이다.

이와는 달리, UChoo는 사건에 가중치를 부여할 수 있으므로 잘 분류가 되는 사건들은 적은 가중치를 부여하여 분류기에 포함시킴으로써, 정보의 누락이 줄어들어 분류 성능을 높일 수 있다. 위의 예제를, 확장된 데이터 표현에 적용해보면 해당 노드에 클래스 1은 0.7, 클래스 2는 0.3이란 확률을 할당해 줄 수 있다.

즉, 말단 노드에서 완전히 분류되지 않고 여러 클래스가 혼재되어있는 상황에서, 가장 높은 확률을 가지는 클래스를 대표 클래스로 선택하는 것 대신에, 각각의 클래스가 가지는 확률 정보를 모두 학습 알고리즘에서 고려하기 때문에 가장 높은 확률을 가지는 클래스가 선택되고 나머지 정보를 버리는 일반적인 학습 방법에 비해 정보를 잃는 것을 피할 수 있다. 따라서 UChoo 알고리즘은 실제 정보를 정확하게 반영할 수 있다는 장점이 있다.

이러한 확률적 결과 표현을 사용하는 UChoo를 부스팅에 적용한 UChoo boost에 비해, Learn++는 결정적 가중치 투표방식을 사용한다. 이에 따라 나타나는 성능의 차이를 실험에서 나타내었으며 알고리즘은 그림 3과 같다.

먼저 해당 데이터베이스(D)에 있는 모든 사건들의 가중치(w)를 동일하게 초기화한다.

각 사건의 가중치는 데이터베이스 안의 모든 사건들의 가중치의 합으로 다음 식 (1)과 같이 정규화(Normalization)한다. 두 번째 차례부터 각 사건의 가중치가 변해 가중치의 총 합도 변한다.

$$D_i^t = w_i^t \Big/ \sum_{j=1}^{n} w_j^t \tag{1}$$

가중치에 따라 데이터베이스 안의 사건들로부터 훈련 사건 집합을 뽑는다. 각 차례에서 UChoo Learner는 훈련 사건 집합을 학습하여 약한 분류기는 가설 (2)을 얻는다.

$$h^t = \left\{ h^t(x_i) | x_i \in DB_m \right\} \tag{2}$$

훈련 사건 집합의 확장된 데이터 표현에서 가중치(1)는 가설 (2)을 정의하는데 사용된다.

$$e^t = \sum_{i:\ \underset{k}{\arg\max}\, h_j^t(x_i) \neq \underset{k}{\arg\max}\, y^k(x_i)} D_i^t \tag{3}$$

약한 분류기들에서 가설 (2)의 에러 (3)를 계산한 뒤 합하여 강한 분류기를 만든다.

$$H^t(x_i) = \frac{\sum_{j=1}^{t} \log(1 - e^j \Big/ e^j) h^j(x_i)}{K} \tag{4}$$

강한 분류기의 에러 (4)를 계산하여 잘못 측정된 사건일 경우 가중치를 증가시켜준다. 다음 차례에서 변화된 가중치를 통해 훈련 사건 집합을 다시 뽑을때는 예측하기 어려웠던 사건이 더 많이 포함된다. 따라서 분류기가 해당 사건을 더 중요하게 다룰 수 있게 된다.

가설들의 합성은 앙상블 기반 알고리즘에서 매우 중요하다. 본 논문에서는 확장된 데이터 표현을 적용하여 가설들을 합성할 때, 가중치를 사용하여 대표 클래스만

이 아닌 모든 클래스를 포함한다.

$Do\ for$ each data base $DB_m, m = 1, M$

$Initialize\ w_i^1 = D_i^1 = \frac{1}{n}$

$for\ t = 1, T$

1. $D_i^t = \frac{w_i^t}{\sum_{j=1}^{n} w_j^t}$
2. Choose training subset TR_t according to D_t
3. Train UChooLearn with TR_t and obtain hypothesis $h^t = \{h^t(x_i) | x_i \in DB_m\}$
4. Compute error of h^t:

$$e^t = \sum_{i:\arg\max_k h_j^t(x_i) \neq \arg\max_k y^k(x_i)} D_i^t$$

($y^k(x_i)$ is the probability of the k_{th} class for a given x_i)

Compound error of

$$H^t(x_i) = \frac{\sum_{j=1}^{t} \log\left(\frac{1-e^j}{e^j}\right) h^j(x_i)}{K}, x_i \in DB_m$$

5. Compute error of H^t:

$$E^t = \sum_{i:\arg\max_j H_j^t(x_i) \neq \arg\max_j y^j(x_i)} D_i^t$$

6. Update weight:

$$w_i^{t+1} = \begin{cases} \frac{E^t}{1-E^t} \cdot w_i^t, \text{if}(\arg\max_j H_j^t(x_i) \neq \arg\max_j y^j(x_i)) \\ w_i^t, \text{otherwise} \end{cases}$$

$end\ for$

$end\ Do$

$$H_m^{final}(x_i) = \frac{\sum_{m=1}^{M} \sum_{j=1}^{T} \log\left(\frac{1-E^j}{E^j}\right) H_m^j(x_i)}{K}$$

$$H^{final}(x) = \arg\max_{c_i} H_i^{final}$$

그림 3. UChoo boost 알고리즘
Fig. 3. UChoo boost algorithm

Ⅲ. 실 험

BEWS는 트레이닝과 테스트 과정을 포함한다. 트레이닝에서는 깨끗한 물과 오염된 물에서 데이터를 얻었다. 그리고 물고기가 있는 어항에 적당량의 독을 넣고 그 행동을 관찰한다. 여기서 얻어진 데이터를 이용하여 분류기를 만든다. 테스트에서는 모니터링 장치로부터 실시간에 입력된 데이터를 트레이닝 프로세스에서 만들어진 규칙에 입력하여 결과 값을 얻는다.

아래 그림 4은 본 실험을 위해 구현된 BEWS의 화면이다.

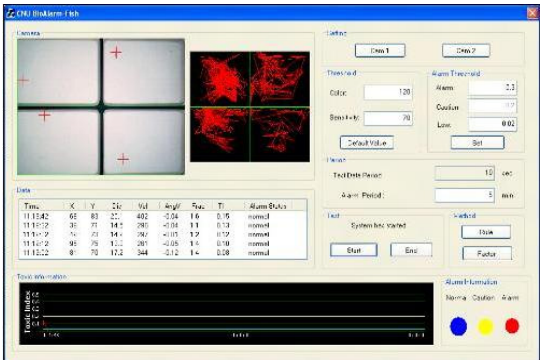

그림 4. 애플리케이션에서의 데이터 수집
Fig. 4. Data capture in the application

변수 선택은 분류하는 애플리케이션에서 중요하다. 의사결정트리에서 적절한 변수를 사용하면 분류하는 애플리케이션에서 의사결정트리의 트레이닝 시간을 줄일 수 있고 분류 수행도 많이 향상시킬 수 있다. 특성 변수 선택에는 물고기의 행동 특징을 사용하였다. 애플리케이션 환경에서는 물고기가 매우 빠르게 움직이고 있으므로 시간 T 동안의 평균값을 사용하였다. 시간 T 동안의 물고기의 행동 특징은 프랙탈 차원뿐만 아니라 x좌표, y좌표, 거리, 절대 거리, 각도 변수로 정의되었다[5].

먼저, 데이터에 10-fold cross validation 를 적용하였다. 데이터를 열 개의 블럭으로 나누고, 그 중 한 개의 블럭으로 테스트 데이터로 정하고, 나머지 아홉 개의 블럭들로 트레이닝 데이터를 만들었다. 분류기의 정확도는 열 개의 블럭들이 모두 테스트 될 수 있도록 10번의 테스트를 위와 같이 한 뒤, 그 평균값으로 측정한다.

생물적 데이터에는 이상 행동을 보이는 개체에 의한 노이즈가 발생할 수 있다. 이는 분류기로 하여금 잘못된 판단을 내리게 하여 성능에 악영향을 줄 수 있다. 이를 해결하기 위해 데이터의 평균값을 사용하여 노이즈의 영향을 줄일 수 있다. 평균값 10은 사건 값을 적용할 때 해당 사건 전후 10개의 평균값을 적용한다는 의미다. BEWS에서 평균값을 적용한 실험 결과는 일정수치까지 보다 많은 데이터의 평균값을 적용하였을 때 오차를 줄여주는데 성공적이었음이 입증되었다[12].

표 2. BEWS에 UChoo Boost를 적용한 실험 결과
Table. 2 Experimental result of BEWS with UChoo Boost

	Error rate(%)		
Average value	UChoo	Learn++	UChoo Boost
1	0.311070	0.264000	0.301667
2	0.261665	0.240250	0.233670
3	0.245197	0.232500	0.224330
4	0.252516	0.218420	0.219250
5	0.236048	0.211250	0.199000
6	0.215919	0.202500	0.193920
7	0.212260	0.203000	0.193670
8	0.203111	0.196833	0.192250
9	0.222324	0.180080	0.180250
10	0.201281	0.177667	0.173330
20	0.164684	0.133083	0.130750
30	0.127173	0.115083	0.109750
50	0.107045	0.086917	0.080830
100	0.058500	0.058500	0.056830
1000	0.016468	0.012833	0.010580

본 논문에서는 이 평균값이 된 데이터를 가지고 Learn++ 알고리즘과 UChoo Boost를 BEWS의 분류기로 적용하여 오차율을 측정해 보았다. 표 2는 각각 평균값이 된 데이터를 UChoo와 Learn++, UChoo Boost 알고리즘을 적용하여 오차율을 측정한 결과이다.

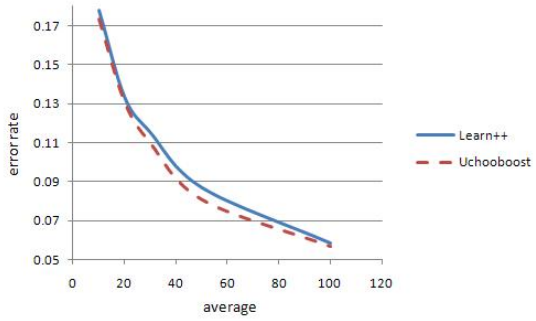

그림 5. Learn++ 와 UChoo Boost의 에러율 비교
Fig. 5. Comparison of Errors in Learn++ and UChoo Boost

실험결과 표 2에서 점진적 학습법인 Learn++와 UChoo boost를 적용한 BEWS의 성능이 하나의 의사결정 분류기를 갖는 일반적인 학습법인 UChoo를 적용한 BEWS[12]보다 뛰어난 것을 확인하였다. 그리고 표 2의 Learn++와 UChoo boost의 실험결과를 그래프로 나타낸 그림 4에서 Learn++보다 UChoo boost를 적용하였을 때 대체적으로 성능이 더 뛰어난 것을 확인할 수 있다.

Ⅳ. 결 론

본 논문에서는 BEWS에 확률적 가중치를 이용한 과반수 투표방식을 적용하여 데이터에 있는 정보를 보다 충실히 나타낼 수 있도록 하는 방법을 제안하였다.

물은 인간을 포함한 모든 생명활동에 중요한 부분이며 짧은 시간 및 적은 이상으로도 큰 문제를 발생 시킬 수 있기에 수질 측정에 대한 신뢰성을 높이기 위한 연구가 계속되어 왔다. BEWS는 기존의 화학적인 측정 방법의 단점이었던 실시간 및 정의하지 않은 물질에 대한 검사가 가능했지만, 생물을 이용한 데이터의 특성상 정의하기 힘든 개체에 대한 판단과 이상행동을 보이는 개체에 의한 영향 등을 해결해야 하는 문제가 있었다. 본 연구로 문제점을 개선하였으나 시스템의 판단에 대한 신뢰도를 보다 높이려면 다양한 분류기법과 부스팅 기법의 적용에 관한 지속적인 연구가 필요하다.

참고문헌

[1] B. W. Scharf, A fish test alarm device for the continual recording of acute toxic substances in water, Arch. Hydrobiol. 85, 250 - 256, 1979

[2] P. Schmitz, F. Krebs, U. Urmer, Development, testing and implementation of automated biotests for the monitoring of the Rhine River, demonstrated by bacteria and algae tests. Water Sci.Technol. 29 (3), pp.215 - 221, 1994.

[3] W. H. ver der Schalie, T. R. Shedd, P. L. Knechtges, M. W. Widder, “Using higher organisms in biological early warning systems for real-time toxicity detection”, 2001 Elsevier Science B.V. 0956-5663/01.
[4] Y. R. Li, D. H. Seo, W. D. Lee, “A New Classifier Applied to Biological Early Warning Systems for Toxicity Detection” ICADIWT 2008, pp.360-365, August 2008.
[5] Y. R. Li, D. H. Seo, W. D. Lee “A New Classification Application of Biological Early Warning Systems for Toxicity Detection” CSA2008, pp.239-242, October 2008.
[6] R. Polikar, “Bootstrap-Inspired Techniques in Computational Intelligence”, IEEE Signal Processing Magazine, pp.59-72, July 2007.
[7] P. N. Tan, M. SteinBach, V. Kumar, “Introduction to data mining”, 2005
[8] Y. Freund, R. Schapire, “Decision-theoretic generalization of on-line learning and an application to boosting,” J. Comput. Syst. Sci. vol.55, no.1, pp.119 - 139, 1997.
[9] D. H. Kim, D. H. Lee, W. D. Lee, “Classifier using Extended Data Expression”, IEEE Mountain Workshop on Adaptive and Learning Systems. pp.154-159, July 2006.
[10] M. J. Turner, J. M. Blackledge, P. R. Andrews, “Fractal Geometry in Digital Imaging”, 1998
[11] J. Wu, Y. S. Kim, C. H. Song, W. D. Lee, ”A New Classifier to Deal with Incomplete Data” ACIS-SNPD2008, 105-110, August 2008.
[12] S. Y. Kim, K. Y. Kwon, W. D. Lee, ”A Biological Early Warning System for Toxicity Detection” NCM2009, pp.1157-1160, August 2009.

저자소개

김성용(Sung Yong Kim)

2009년 충남대학교 전기정보통신공학부 컴퓨터전공(학사)
2009년~현재 충남대학교 컴퓨터공학과 석사과정
※관심분야 : 인공지능, 멀티미디어, 데이터마이닝

권기용(Ki Yong Kwon)

2009년 충남대학교 전기정보통신공학부 컴퓨터전공(학사)
2009년~현재 충남대학교 컴퓨터공학과 석사과정
※관심분야 : 인공지능, 멀티미디어, 데이터마이닝

이원돈(Won Don Lee)

1979년 서울대학교(석사)
1982년 U.of Illinois 대학원(석사)
1986년 U.of Illinois 대학원(박사)
1987년~ 현재: 충남대학교 컴퓨터공학과 교수
※관심분야 : 신경회로망, 멀티미디어, 데이터마이닝